2.5 Main advantage of the layered approach to system design

Modularity & abstraction: Each layer is built on top of lower layers, and each only uses services of the layer directly beneath it.

→ This makes debugging, testing, and system design easier because changes in one layer don’t directly affect others.

Example: If the CPU scheduling algorithm changes (in the processor management layer), it won’t affect the file system or user interface layers.

Disadvantages of the layered approach

Performance overhead: Communication between layers adds overhead because requests must pass through multiple interfaces.

Rigid structure: A strict layered approach may force designers to fit functionality into a specific layer, even if it doesn’t naturally belong there.

Design complexity: Determining the exact number and responsibilities of layers can be hard.

2.6 Five services provided by an operating system

Program execution

OS loads programs into memory and runs them.

Convenience: Users don’t need to manually manage loading, starting, or terminating programs.

Impossible at user level: Programs cannot run other programs directly without OS support.

File system manipulation

OS provides access to files and directories (create, read, write, delete).

Convenience: Users don’t need to handle raw disk sectors.

Impossible at user level: User programs can’t directly control disk hardware safely.

I/O operations

OS manages input/output devices (keyboard, display, printer, etc.).

Convenience: Programs use simple I/O commands instead of controlling hardware.

Impossible at user level: Devices require privileged instructions not accessible to user programs.

Communication (interprocess communication and networking)

OS allows processes to exchange information (shared memory, message passing, sockets).

Convenience: Users don’t have to design synchronization/communication mechanisms from scratch.

Impossible at user level: Without OS, direct process-to-process communication would be unreliable and insecure.

Error detection & handling

OS monitors system resources, detects errors (bad memory, disk failure, illegal instructions), and responds.

Convenience: Prevents crashes and data corruption.

Impossible at user level: User programs cannot monitor hardware errors globally.

2.7 Why do some systems store the operating system in firmware (ROM/EPROM)?

Reasons:

Embedded systems & appliances (e.g., routers, microwaves): OS must always be available and doesn’t change often.

Boot reliability: OS in ROM ensures the system can start without loading from external storage.

Security: Firmware is harder to tamper with compared to disk-based OS.

Downside:

Hard to upgrade or patch (requires flashing ROM).

Limited storage size compared to disk-based OS.